AMENDMENTS TO THE CLAIMS

1   1.   (Currently Amended) A document-generation process performed using a

2 computer system, the process comprising:

3 parsing a raw document to create an internal representation of the document;

4   reading a <u>first</u> first-level transform from a transform database;

5   applying the <u>first</u> first-level transform to the internal representation ~~so as~~ to create

6   a <u>first</u> first-level document;

7   writing the <u>first</u> first-level document to memory;

8   <u>reading a second first-level transform from the transform database;</u>

9   <u>applying the second first-level transform to the internal representation to create a</u>

10   <u>second first-level document;</u>

11   receiving a first request for a second-level document that depends from the <u>first</u>

12   first-level document;

13   in response to the first request, reading a second-level transform from the

14   transform database;

15   in response to the first request, applying the second-level transform to the <u>first</u>

16   first-level document ~~so as~~ to create [[a]] <u>the</u> second-level document; ~~and~~

17   writing the second-level document to memory<u>; and</u>

18   <u>writing the second first-level document to memory;</u>

19   <u>wherein the first first-level document and the second first-level document are</u>

20   <u>different.</u>

1   2.   (Currently Amended) A document-generation process as defined in Claim

2 1, further comprising:

3   revising the raw document;

4   applying the <u>first</u> first-level transform to the revised raw document ~~so as~~ to create

5   a revised <u>first</u> first-level document;

6   writing the revised <u>first</u> first-level document to memory; and

7   indicating that a then-existing second-level document is invalid.

1   3.   (Original)   A document-generation process as defined in Claim 2,
2   further comprising:
3   receiving a second request for the then-existing second-level document;
4   determining that the then-existing second-level document has been indicated
5   invalid;
6   applying the second-level transform to the revised first first-level document ~~so as~~
7   to create a revised second-level document; and
8   writing the revised second-level document to memory.

1   4.   (Currently Amended) A document-generation process as defined in Claim
2   1, further comprising:
3   receiving a request for a revised first first-level transform;
4   revising the then-existing first first-level transform;
5   applying the revised first first-level transform to the raw document ~~so as~~ to create
6   a revised first first-level document;
7   writing the revised first first-level document to memory; and
8   indicating that the then-existing second-level document is invalid.

1   5.   (Currently Amended) A document-generation process as defined in Claim
2   4, further comprising:
3   receiving a second request for the then-existing second-level document;
4   determining that the then-existing second-level document has been indicated
5   invalid;
6   applying the second-level transform to the revised first first-level document ~~so as~~
7   to create a revised second-level document; and
8   writing the revised second-level document to memory.

1   6.   (Original)   A document-generation process as defined in Claim 1,
2   wherein a respective GID is assigned to each of the first-level document and the second-
3   level document.

1   7.   (Currently Amended) A document-generation process as defined in Claim

2   1, wherein the first first-level document and the second-level document are timestamped

3   and wherein a document is deleted when a timestamp indicates that the document is stale.

1   8.   (Currently Amended) A method of generating customized versions of a

2   document using a computer system, the method comprising:

3   storing the document in raw form;

4   parsing the document to create an internal representation of the document;

5   receiving a request to generate a second-level document; and

6   decomposing the document to create the second-level document, wherein

7   decomposing the document comprises:

8   applying a first first-level transform to the internal representation of the

9   document so as to create a first first-level document; [[and,]] and

10   in response to the request to generate the second-level document, applying

11   a second-level transform to the first first-level document so as to

12   create the second-level document; and

13   decomposing the document to create a second first-level document, wherein

14   decomposing the document to create the second first-level document

15   comprises:

16   applying a second first-level transform to the internal representation of the

17   document to create a second first-level document;

18   wherein the first first-level document and the second first-level document are

19   different.

1   9.   (Currently Amended) A method as defined in Claim 8, wherein applying

2   a first first-level transform and applying a second-level transform comprises applying

3   sequential transforms to the document.

1   10.   (Canceled)

1   11.   (Canceled)

1   12.   (Currently Amended) A method as defined in Claim 8, wherein
2 decomposing the document comprises applying a third-level transform to the second
3 level document ~~so as~~ to create a third-level document.

1   13.   (Original)   A method as defined in Claim 8, wherein the document is
2 stored in raw XML form.

1   14.   (Currently Amended) A method as defined in Claim 13, wherein applying
2 a <u>first</u> first-level transform and applying a second-level transform comprises applying
3 sequential transforms to the document.

1   15.   (Currently Amended) A method as defined in Claim 14, wherein applying
2 a <u>first</u> first-level transform of the document stored in raw XML form comprises applying
3 a subscription-level transform to the internal representation of the document ~~so as~~ to
4 create a subscription-level document.

1   16.   (Previously Presented)   A method as defined in Claim 15, wherein
2 the subscription-level transform enables content filtering of the internal representation in
3 accordance with a user's request.

1   17.   (Currently Amended) A method as defined in Claim 16, wherein applying
2 a second-level transform comprises applying an organization-level transform to the
3 subscription-level document ~~so as~~ to create an organization-level document.

1   18.   (Original)   A method as defined in Claim 17, wherein the
2 organization-level transform enables subscription-specific content filtering of a
3 subscription-level document.

1   19.   (Currently Amended) A method as defined in Claim 18, wherein
2 decomposing the document comprises applying a presentation-level transform to the
3 organization-level document ~~so as~~ to create a presentation-level document.

1   20.   (Original)   A method as defined in Claim 19, wherein the presentation-

2   level transform generates an organization-specific document for end user presentation.

1   21.   (Original)   A method as defined in Claim 20, wherein the presentation-

2   level transform generates an HTML document or a text file for end user presentation.

3   22.   (Original)   A method as defined in Claim 21, wherein the subscription-

4   level transform is mandatory and the organization-level and presentation-level transforms

5   are optional.

1   23.   (Original)   A method as defined in Claim 8, wherein a transform is

2   applied to a document only as a result of an initial demand for a transformed document.

1   24.   (Original)   A method as defined in Claim 23, wherein the demand for a

2   transformed document is a client request.

1   25.   (Original)   A method as defined in Claim 24, wherein the demand for a

2   transformed document is a document publication process.

1   26.   (Original)   A method as defined in Claim 23, wherein transformed

2   documents are written to a cache.

1   27.   (Original)   A method as defined in Claim 26, wherein demands for a

2   transformed document, subsequent to the initial demand, are referred to the cache.

1   28.   (Currently Amended) A computer readable medium having data stored

2   therein to cause a data processing system to generate a data document according to a

3   process comprising:

4   storing a raw form of the document;

5   parsing the document to create an internal representation of the document; and

6   receiving a request from a client computer system coupled to the data processing

7   system to generate a second-level document into a particular form;

8   decomposing the document into the form requested by the client system, wherein

9   decomposing the document comprises:

10   applying a <u>first</u> first-level transform to the internal representation of the

11   document ~~so as~~ to create a <u>first</u> first-level document; [[and,]] <u>and</u>

12   in response to the request to generate the second-level document, applying

13   a second-level transform to the <u>first</u> first-level document ~~so as~~ to

14   create the second-level document; <u>and</u>

15   <u>decomposing the document to create a second first-level document, wherein</u>

16   <u>decomposing the document to create the second first-level document</u>

17   <u>comprises:</u>

18   <u>applying a second first-level transform to the internal representation of the</u>

19   <u>document to create a second first-level document;</u>

20   <u>wherein the first first-level document and the second first-level document are</u>

21   <u>different.</u>

1   29.   (Currently Amended) A computer readable medium as defined in Claim

2 28, wherein applying a <u>first</u> first-level transform and applying a second-level transform

3 comprises applying sequential transforms to the document.

1   30.   (Previously Presented)   A computer readable medium as defined in

2 Claim 28, wherein the document is stored in XML form.

1   31.   (Previously Presented)   A computer readable medium as defined in

2 Claim 30, wherein the document stored in XML form is parsed by an XML parser to

3 create the internal representation.

1   32.   (Previously Presented)   A computer readable medium as defined in

2 Claim 31, wherein the internal representation level of the document is transformed to a

3 subscription-level document by applying a subscription-level transform to the internal

4 representation.

1   33.   (Currently Amended) A computer readable medium as defined in Claim
2   32, wherein application of the subscription level transform to the internal representation
3   ~~so as~~ to create a subscription-level document is required.

1   34.   (Previously Presented)   A computer readable medium as defined in
2   Claim 32, wherein the subscription-level document is transformed into an organization-
3   level document by applying an organization-level transform to the subscription-level
4   document.

1   35.   (Previously Presented)   A computer readable medium as defined in
2   Claim 34, wherein application of the organization-level transform to the subscription-
3   level document ~~so as~~ to create an organization-level document is optional.

1   36.   (Previously Presented)   A computer readable medium as defined in
2   Claim 34, wherein the internal representation of the document is decomposed to a
3   transform-level document only in response to a request for a transform-level document.

1   37.   (Previously Presented)   A computer readable medium as defined in
2   Claim 36, wherein transformed documents are written to a cache.

1   38.   (Previously Presented)   A computer readable medium as defined in
2   Claim 37, wherein an initial request for a transformed document causes decomposition of
3   the internal representation into the form requested and wherein subsequent requests for a
4   transformed document causes the transformed document to be retrieved from memory.

1   39.   (Previously Presented)   A computer readable medium as defined in
2   Claim 29, wherein the data document is generated according to a process comprising:
3   tracking the dependencies of a transformed document; and
4   regenerating the transformed document when any dependency related to the
5   document changes.

1   40.   (Previously Presented)   A computer readable medium as defined in

2   Claim 39, wherein the document is generated according to a process comprising:

3   designating a cached version of the document invalid when any dependency

4   related to the document changes, and

5   regenerating the transformed document in response to a request for the document

6   that is made after the dependency change.

1   41.   (Previously Presented)   A computer readable medium as defined in

2   Claim 40, wherein the document is stored in XML form.

1   42.   (Previously Presented)   A computer readable medium as defined in

2   Claim 39, wherein the document stored in XML form is parsed by an XML parser to

3   create the internal representation.

1   43.   (Previously Presented)   A computer readable medium as defined in

2   Claim 42, wherein the internal representation level of the document is transformed to a

3   subscription-level document by applying a subscription-level transform to the internal

4   representation.

1   44.   (Previously Presented)   A computer readable medium as defined in

2   Claim 43, wherein application of the subscription level transformed to the internal

3   representation ~~so as~~ to create a subscription-level document is required.

1   45.   (Previously Presented)   A computer readable medium as defined in

2   Claim 43, wherein the subscription-level document is transformed into an organization-

3   level document by applying an organization-level transform to the subscription-level

4   document.

1   46.   (Previously Presented)   A computer readable medium as defined in

2   Claim 45, wherein application of the organization-level transform to the subscription-

3   level document ~~so as~~ to create an organization-level document is optional.

1   47. (Previously Presented) A computer readable medium as defined in

2 Claim 45, wherein the internal representation of the document is decomposed to a

3 transform-level document only in response to a request for a transform-level document.

1   48. (Previously Presented) A computer readable medium as defined in

2 Claim 47, wherein transformed documents are written to a cache.

1   49. (Canceled)

1   50. (Canceled)

1   51. (Canceled)

1   52. (Canceled)

1   53. (Currently Amended) A method of generating customized versions of a

2 document using a computer system, the method comprising:

3   storing the document in a primitive form;

4   transforming the document from primitive form into an internal representation of

5   the document;

6   transforming the internal representation into at least ~~one subscription-level~~ a first

7   first-level document[[,]] and into a ~~DEFAULT organization-level~~ second

8   first-level document ~~and into at least one user-specific organization-level~~

9   ~~document~~;

10   transforming the ~~DEFAULT organization-level~~ first first-level document into at

11   least one ~~presentation-level~~ first second-level document; and

12   transforming the ~~user-specific organization-level~~ second second-level document

13   into at least one ~~presentation-level~~ second second-level document;

14   wherein the first first-level document and the second first-level document are

15   different.

1   54.   (Currently Amended) A method of generating customized versions of

2 documents as defined in Claim [[53]] 71, wherein the user-specific organization-level

3 document is transformed into at least two presentation-level documents.

1   55.   (Original)   A method of generating customized versions of documents

2 as defined in Claim 54, wherein the user-specific organization-level document is

3 transformed into an HTML presentation-level document and into a FLAT presentation-

4 level transform.

1   56.   (Currently Amended) A method of generating customized versions of a

2 document as defined in Claim [[53]] 71, wherein:

3   (i) the internal representation is transformed into a first subscription-level

4   document and into a second subscription-level document;

5   (ii) the first subscription level document is transformed into a subscription-level

6   specific DEFAULT organization-level document and into at least one

7   user-specific organization-level document; and

8   (iii) the second organization-level document is transformed into a subscription-

9   level-specific DEFAULT organization-level document.

1   57.   (Original)   A method of generating customized versions of a document

2 as defined in Claim 56, wherein the document is stored in XML form.

1   58.   (Currently Amended) A system for the generation of customized data

2 documents, the system comprising:

3   first database for storing raw data documents;

4   first tabular means for storing document records;

5   an interface coupling the first database to the first tabular means;

6   a request interface to receive a document generation request, wherein the

7   document generation request indicates a particular document type;

8   a second database for storing a compilation of transforms that enable an internal

9   representation of a document to be transformed into a first first-level

10	document <u>and into a second first-level document</u> and that enable the <u>first</u>
11	first-level document to be transformed into a second-level document,
12	wherein each transform is mapped to a particular document type;
13	a document generator, coupled to the first and second databases and first and
14	second tabular means, and to the request interface, to generate the <u>first</u>
15	~~first-level~~ <u>first-level</u> document using at least one of the transforms<u>, to</u>
16	<u>generate the second first-level document using at least one of the</u>
17	<u>transforms that is different than at least one of the transforms used to</u>
18	<u>generate the first-level document,</u> and to generate the second-level
19	document using at least one of the transforms in response to receipt of the
20	document generation request, wherein the document type of the second-
21	level document and the transform used to generate the second-level
22	document are indicated by the document generation request;
23	second tabular means for storing transform records; and
24	an interface coupling the second database to the second tabular means<u>; and</u>
25	<u>wherein the first first-level document and the second first-level document are</u>
26	<u>different.</u>

1	59. (Original) A system as defined in Claim 58, further comprising:
2	a cache coupled to the first tabular means and to the second tabular means.

1	60. (Currently Amended) A system as defined in Claim 58, wherein the
2	second database stores a compilation of transforms that enable:
3	transforming an internal representation into at <u>least</u> one subscription-level
4	document, into a DEFAULT organization-level document and into at least
5	one user-specific organization-level document;
6	transforming the DEFAULT organization-level document into at least one
7	presentation-level document; and
8	transforming the user-specific organization-level document into at least one
9	presentation-level document.

1   61.   (Previously Presented)   A system as defined in Claim 60, wherein
2   the second database stores a compilation of transforms that enable:
3   (i) the internal representation to be transformed into a first subscription-level
4   document and into a second subscription-level document;
5   (ii) the first subscription level document to be transformed into a subscription-
6   level specific DEFAULT organization-level document and into at least
7   one user-specific organization-level document; and
8   (iii) the second organization-level document to be transformed into a subscription-
9   level-specific DEFAULT organization-level document.

1   62.   (Original)   A system as defined in Claim 61, further comprising:
2   a cache coupled to the first tabular means and to the second tabular means.

1   63.   (Previously Presented) A document-generation process as defined in
2   Claim 1 wherein receiving a first request for a second-level document that depends from
3   the first-level document comprises:
4   receiving a first request from a client computer coupled to the computer system
5   for a second-level document that depends from the first-level document

1   64.   (Currently Amended) A document-generation process as defined in Claim
2   1 further comprising:
3   tracking the dependencies of a transformed document, wherein the transformed
4   document includes at least one [[or]] of the first first-level document and
5   the second-level document and the dependencies include the first first-
6   level document and the internal representation; and
7   regenerating the transformed document when any dependency related to the
8   document changes.

1   65.   (Previously Presented) A method of generating customized versions of a
2   document using a computer system as defined in Claim 8 wherein receiving a request to
3   generate a second-level document comprises:

4   receiving a request from a client computer coupled to the computer system to

5   generate a second-level document.

1   66.   (Currently Amended) A method of generating customized versions of a

2   document using a computer system as defined in Claim 8 further comprising:

3   tracking the dependencies of a transformed document, wherein the transformed

4   document includes at least one [[or]] of the first first-level document and

5   the second-level document and the dependencies include the first first-

6   level document and the internal representation; and

7   regenerating the second-level document when any dependency related to the

8   transformed document changes.

1   67.   (Currently Amended) A method of generating customized versions of a

2   document using a computer system as defined in Claim [[53]] 71 wherein receiving a

3   request to generate a second-level document comprises:

4   receiving a request from a client computer coupled to the computer system to

5   generate [[a]] the second-level document.

1   68.   (Currently Amended) A method of generating customized versions of a

2   document using a computer system as defined in Claim [[53]] 71 further comprising:

3   tracking the dependencies of the transformed documents, wherein the transformed

4   documents include the subscription-level document, the DEFAULT

5   organization-level document, the user-specific organization-level

6   document, the presentation-level documents, and the dependencies include

7   the DEFAULT organization-level document, the user-specific

8   organization-level document, and the internal representation; and

9   regenerating the second-level document when any dependency related to the

10   transformed document changes.

1   69.   (Currently Amended) A method of generating customized versions of a

2   document using a computer system as defined in Claim [[53]] 71 wherein receiving a

3   request to generate a second-level document comprises:

4   receiving a request from a client computer coupled to the computer system to

5   generate [[a]] <u>the</u> second-level document.

1 70. (Currently Amended) A system for the generation of customized data

2 documents as defined in Claim 58 further comprising:

3   a document dependency tracker and regenerator, coupled to the first and second

4   databases and first and second tabular means, to track the dependencies of

5   a transformed document, wherein the transformed document includes at

6   least one or the first-level document and the second-level document and

7   the dependencies include the first-level document and the internal

8   representation and to regenerate the transformed document when any

9   dependency related to the document changes.

1 71. (New) A method of generating customized versions of documents as in

2 Claim 53, wherein the first-level document comprises a subscription-level document and

3 the second first-level document comprises a DEFAULT organization-level document,

4 wherein:

5   transforming the internal representation further comprises:

6   transforming the internal representation into at least one subscription-level

7   document, into a DEFAULT organization-level document and into

8   at least one user-specific organization-level document;

9   transforming the first first-level document further comprises:

10   transforming the DEFAULT organization-level document into at least one

11   presentation-level document; and

12   the method further comprises:

13   transforming the user-specific organization-level document into at least

14   one presentation-level document.

1 72. (New) A document-generation process performed using a computer

2 system, the process comprising:

3 parsing a raw document to create an internal representation of the document;

4   reading a first first-level transform from a transform database;

5   applying the first first-level transform to the internal representation to create a

6   first first-level document;

7   writing the first first-level document to memory;

8   reading a second first-level transform from the transform database;

9   applying the second first-level transform to the internal representation to create a

10   second first-level document;

11   reading a second-level transform from the transform database;

12   applying the second-level transform to the first first-level document to create a

13   second-level document;

14   writing the second-level document to memory; and

15   writing the second first-level document to memory;

16   wherein the first first-level document and the second first-level document are

17   different.

1   73.   (New) A computer readable medium comprising data stored therein to

2   cause a data processing system to:

3   read a first first-level transform from a transform database;

4   apply the first first-level transform to the internal representation to create a first

5   first-level document;

6   write the first first-level document to memory;

7   read a second first-level transform from the transform database;

8   apply the second first-level transform to the internal representation to create a

9   second first-level document;

10   read a second-level transform from the transform database;

11   apply the second-level transform to the first first-level document to create a

12   second-level document;

13   write the second-level document to memory; and

14   write the second first-level document to memory;

15   wherein the first first-level document and the second first-level document are

16   different.